The gravity9 Delivery Model

A description of how we approach delivery

High-level overview
September 2024

gravity9

Agenda

Our Delivery Model

Delivery Phases

Delivery Tools

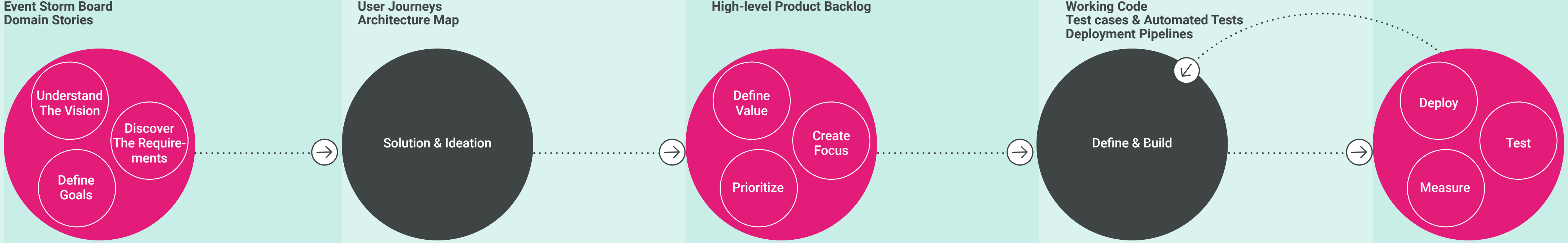
How to get started

Transitioning into solutioning

Summary

Our Delivery Model

Our delivery model defines the flow of how we develop products and implement projects. It starts with understanding current business processes. Based on this understanding we develop high-level solutions that we break into incremental pieces that can be defined in detail, implemented and released. This enables feedback loops and provides opportunity for adjustments if needed.



OUTPUT
Vision Template & Value Curve
Event Storm Board
Domain Stories

1 UNDERSTAND

Understand the opportunity or challenge
Understand how the client currently works

GOAL

What are we trying to accomplish and why?
How do we currently do things?

Tools we use most of the time

- Stakeholder interviews
- Product vision template
- Event storm
- Domain stories - as is
- Value curve
- System analysis
- Standards and reference architecture analysis
- Sub domain definition

Supporting tools (optional)

- Personas
- Secondary research
- North Star metric

Tools we may use

- Competitor analysis
- user journey - as is
- data availability check
- analysis of data infrastructure
- assess regulatory and compliance requirements

Supporting tools (optional)

- User interviews
- Field study
- Evaluate compatibility of technology stack
- Cost-benefit analysis

Typically 2-3 Weeks

OUTPUT
Feature Map
User Journeys
Architecture Map

2 SOLUTION

Define how a product or solution could meet the opportunity or overcome the challenge

GOAL

How could we do things moving forward?

Tools we use most of the time

- Domain stories - to be
- Feature mapping
- User flow
- User journey - to be
- High-level logical architecture diagram
- High-level infrastructure diagram
- Patterns proposal
- Libraries and frameworks proposal

Supporting tools (optional)

- Brand design
- Spikes

Tools we may use

- User interviews
- Field study
- Design sketch
- Risk assessment

Supporting tools (optional)

- Competitor analysis
- Proof of concept
- User tests
- Ethic approval

Typically 2 Weeks

OUTPUT
Product Roadmap
High-level Product Backlog

3 PLAN & PRIORITIZE

Prioritize and plan the solution build

GOAL

In which order do we want to implement our solution?

Tools we use most of the time

- Product roadmap

Supporting tools (optional)

- Value analysis
- Dependency mapping
- Accelerators

Tools we may use

- Card sorting
- Backlog initialization
- Contingency plan

Typically 1-2 Weeks

OUTPUT
Detailed prioritized Product Backlog
Working Code
Test cases & Automated Tests
Deployment Pipelines

4 BUILD

Build out in depth requirements and build the solution incrementally

GOAL

How should the solution work?

Tools we use most of the time

- Product backlog refinement
- Wireframes
- Application domain model definition
- Component suggestion (micro-services, micro-frontends, integrations, identity providers)
- Develop
- Manual qa
- Automated tests
- Devops pipelines

Supporting tools (optional)

- Stakeholder interviews
- User interviews
- Test case creation

Tools we may use

- User observations
- Domain story refinement (to be)
- User journey refinement (to be)
- Feature engineering
- Model selection
- Model integration
- Train and evaluate models
- Establish model registry
- Develop etl pipeline

Supporting tools (optional)

- Data quality check
- Data augmentation
- Check fairness and bais
- Build interoperability & explainability tool
- Data anonymisation
- Data labeling

Typical increment length 1 - 3 Months

OUTPUT
Working Code in Production

5 RELEASE & MEASURE

Release increments of the solution and test them, measure success

GOAL

Does the solution meet user needs?

Tools we use most of the time

- Release support
- Automated deployment

Supporting tools (optional)

- Release planning

Tools we may use

- User observations
- User tests
- Focus groups
- User behavior analytics

Supporting tools (optional)

- Check data & concept drift
- Monitoring models performance

Typically continuously following each BUILD increment

* There could be a return to any step in the framework at any point. If there is a change direction, vision or other variables.

** Timeframe may vary dependent on the nature and complexity of the project, operational considerations such as project resources, budget and other variables.

Delivery Phases

	Understand 1	Solution 2	Plan 3	Build 4	Release 5
WHAT?	Understand the opportunity or challenge broadly AUnderstand technical set up and requirements	Create a solution Broadly define overarching approach Define shared components Define key features	Plan out a path forward Prioritize and define incremental value	Understand the opportunity or challenge in detail Refine requirements and user stories Create wireframes Implement & Test	Release product Gather feedback Assess success criteria results
WHO Is Needed?	Key Stakholders, Business Stakeholders, Technical Stakeholders, Subject Matter Experts	Key Stakeholders, Business Stakeholders, Technical Stakeholders	Key Stakeholders	Key Stakeholders, Subject Matter Experts	Key Stakeholders, Users/Customers
KEY TEAM MEMBERS?	Product Owner, Technical Lead. Architect (depending on the challenge or opportunity)	Product Owner, Product Designer, Technical Lead. Architect	Product Owner, Team Lead	Product Owner, Product Designer, Team Lead, Developers, QA, SDET, Devops	Product Owner, Team Lead
ITERATE?	Ideally once per initiative	Ideally once per initiative	Ongoing	Iterative	Iterative
HOW LONG?	~ 2 weeks	~ 1-2 weeks	~ 1 week	~ 1-3 weeks	Continuous, as needed
HOW?	We use different tools for different scenarios and different challenges, the approach is tailored to the need				
SAMPLE OUTPUTS?	Vision Template & Value Curve Event Storming Board Domain Stories	User Flows Architecture Map Feature Map	Product Roadmap High-level product backlog	Detailed user stories Wireframes Working functionality in lower environment Test cases and automated tests Deployment pipelines	Working functionality in production Documented user feedback

Delivery Phases Explained

1. Understand: Goal is to develop an understanding of the opportunity or challenge at hand and develop a common language. This includes understanding drivers, establishing success criteria and diving into the currently used business processes as well understanding the user base and technology set up.

2. Solution: Goal is to imagine what the future could look like and get buy in from all relevant parties. This phase is exploratory but aims to define a strategic, tangible goal that can be described via an architecture document or a design prototype, or the implementation of a POC depending on the context.

3. Plan: Goal is to define incremental pieces of value that can be implemented iteratively in priority order. The plan will be reflected in a roadmap, milestones are small, preferably around 1-2 months allowing for feedback and course adjustments if needed.

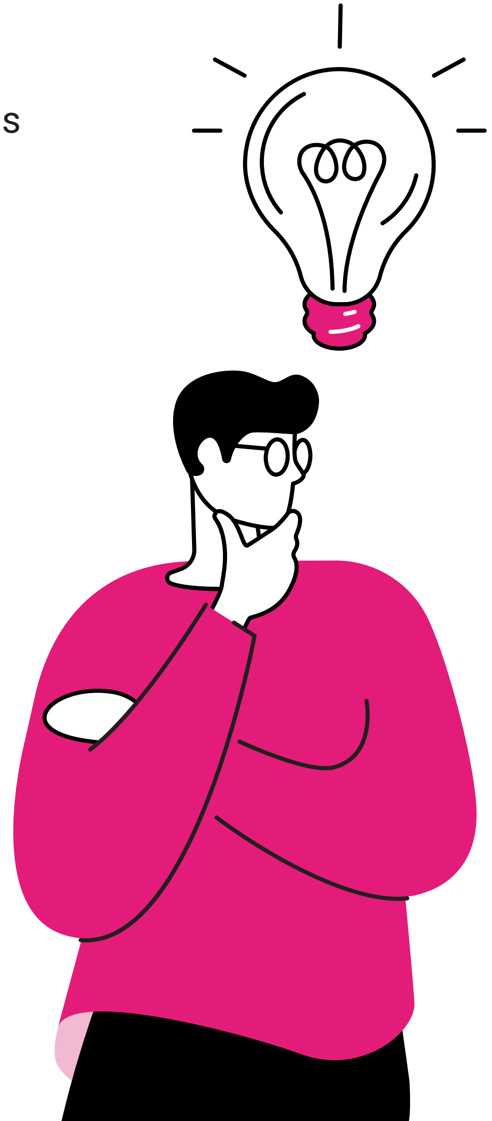
Delivery Phases Continued

4. **Build:** Using a scrum-based agile process we take user stories and build out requirements in detail and implement features and functionality in bi-weekly sprints, following the typically practices of sprint planning, estimations, demos, and retrospectives. Testing is incorporated as part of the sprint. The outcome of each sprint is an incremental piece of functionality towards the larger goal.

5. **Release:** The final step is releasing functionality and features. A release could happen via vertical micro deployments to show value or releasing but hiding features until they are fully assembled. Results are reviewed based on original metrics.

Delivery Tools

One core aspect of our delivery model is tailoring the choice of tools to the business opportunity or challenge at hand. We use some core tools – such as defining and documenting a product vision – in many scenarios. However, other tools are used only if they fit a specific situation and provide value in creating the best product or solution. Tools also vary as they touch different areas of expertise ranging from product to engineering and AI.



1 UNDERSTAND

Understand the opportunity or challenge
Understand how the client currently works

GOAL

What are we trying to accomplish and why?
How do we currently do things?

Tools we use most of the time

- Stakeholder interviews
- Product vision template
- Event storm
- Domain stories - as is
- Value curve
- System analysis
- Standards and reference architecture analysis
- Sub domain definition

Supporting tools (optional)

- Personas
- Secondary research
- North Star metric

Tools we may use

- Competitor analysis
- User journey - as is
- Data availability check
- Analysis of data infrastructure
- Assess regulatory and compliance requirements

Supporting tools (optional)

- User interviews
- Field study
- Evaluate compatibility of technology stack
- Cost-benefit analysis

2 SOLUTION

Define how a product or solution could meet the opportunity or overcome the challenge

GOAL

How could we do things moving forward?

Tools we use most of the time

- Domain stories - to be
- Feature mapping
- User flow
- User journey - to be
- High-level logical architecture diagram
- High-level infrastructure diagram
- Patterns proposal
- Libraries and frameworks proposal

Supporting tools (optional)

- Brand design
- Spikes

Tools we may use

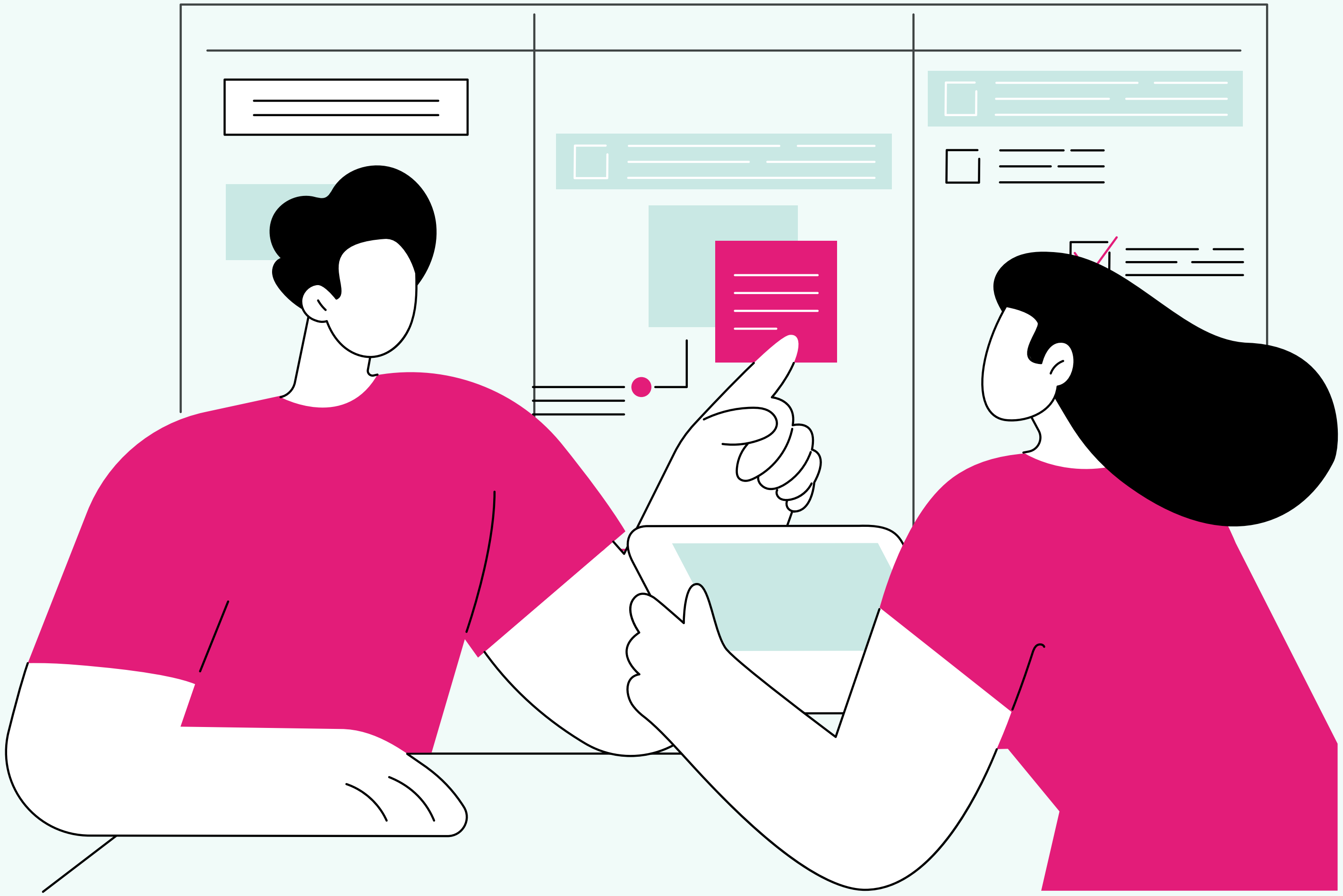
- User interviews
- Field study
- Design sketch
- Risk assessment

Supporting tools (optional)

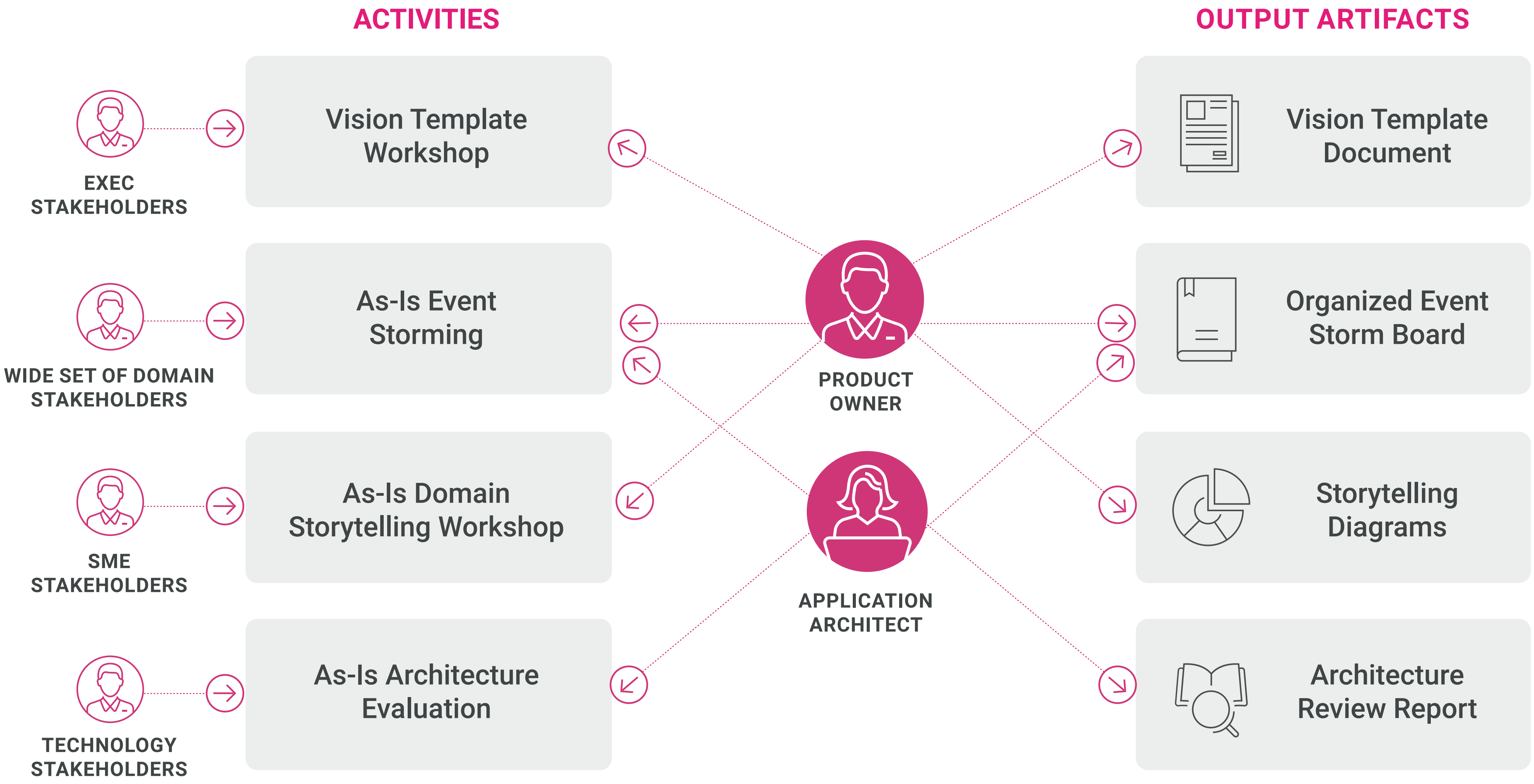
- Competitor analysis
- Proof of concept
- User tests
- Ethic approval

How to get started

Starting a project begins with understanding business workflows. In that context we facilitate different sessions such as a vision template workshop, an event-storming and evaluating business workflows and the current architecture. These activities result in various artifacts such as a vision template, an event storm board, story telling diagrams and an architecture review report. These artifacts help build consensus across stakeholders and create the basis for the solutioning phase.



How to get started



ACTIVITIES

OUTPUT ARTIFACTS

EXEC STAKEHOLDERS

Vision Template Workshop

Vision Template Document

WIDE SET OF DOMAIN STAKEHOLDERS

As-Is Event Storming

Organized Event Storm Board

PRODUCT OWNER

SME STAKEHOLDERS

As-Is Domain Storytelling Workshop

Storytelling Diagrams

APPLICATION ARCHITECT

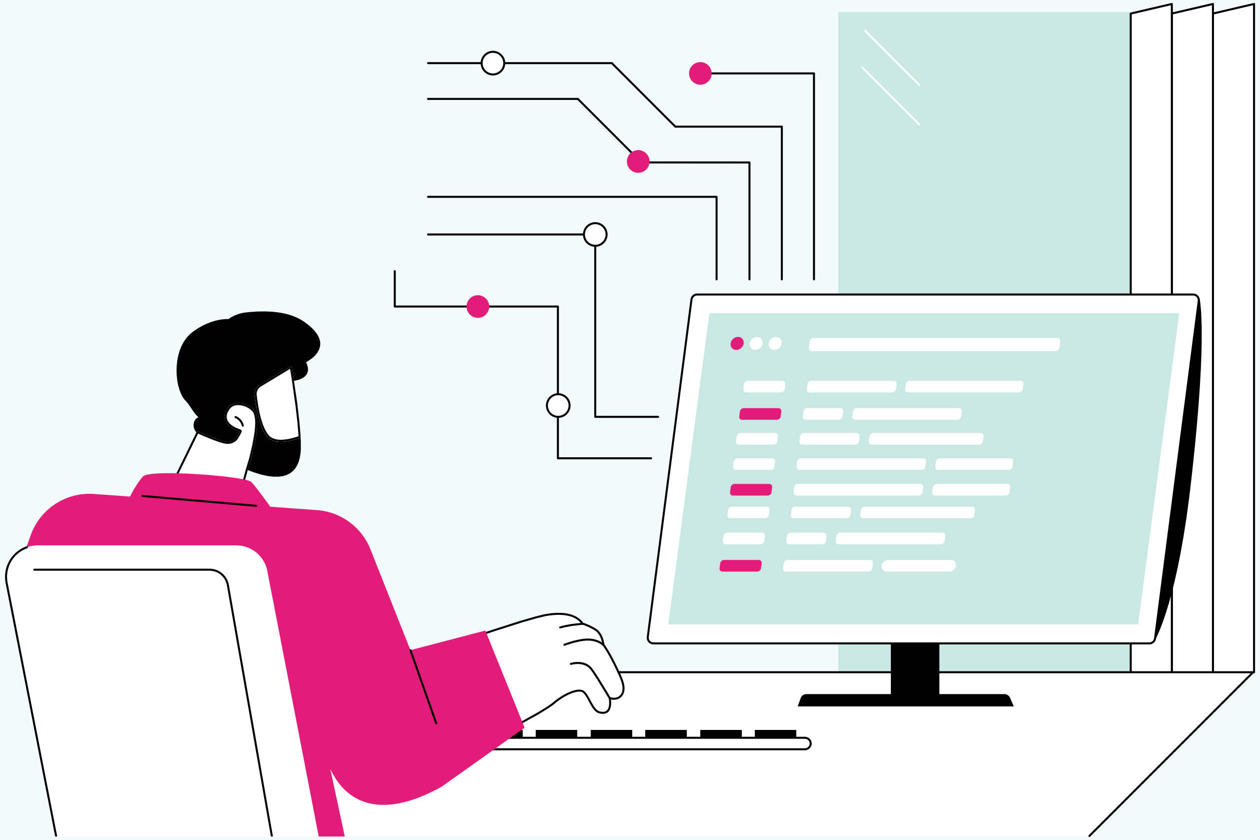
TECHNOLOGY STAKEHOLDERS

As-Is Architecture Evaluation

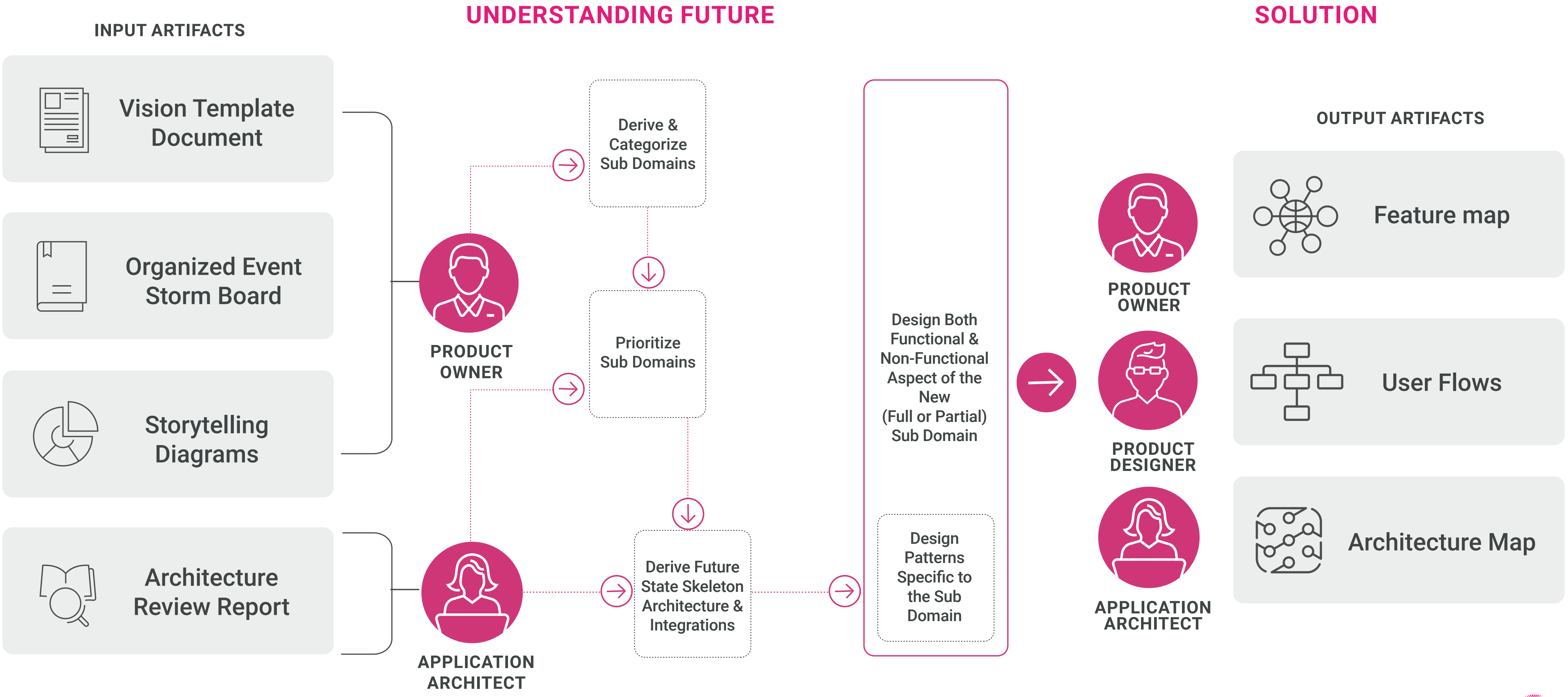
Architecture Review Report

Transition into solutioning

The outputs of the initial workshops enable us to start paving the way for a future solution. This typically happens by deriving and categorizing sub-domains. These sub domains may look different today from what they will look in the future. They build the basis to sketch out a full future solution. As part of this process, we use feature maps, user flows and architecture maps to define the solution from different perspectives. Once we have a solution, we roadmap a path forward and incrementally build the product.



Transition into solutioning



INPUT ARTIFACTS

- Vision Template Document
- Organized Event Storm Board
- Storytelling Diagrams
- Architecture Review Report

UNDERSTANDING FUTURE

PRODUCT OWNER

Derive & Categorize Sub Domains

Prioritize Sub Domains

APPLICATION ARCHITECT

Derive Future State Skeleton Architecture & Integrations

Design Both Functional & Non-Functional Aspect of the New (Full or Partial) Sub Domain

Design Patterns Specific to the Sub Domain

SOLUTION

PRODUCT OWNER

PRODUCT DESIGNER

APPLICATION ARCHITECT

OUTPUT ARTIFACTS

- Feature map
- User Flows
- Architecture Map

Summary

Our delivery model is our approach to delivering products and projects. As our delivery approach is agile so is this delivery model. We expand our toolset and finetune it as we encounter new challenges. It is to be used as a guide towards building amazing things.

“The most certain way to succeed is always to try just one more time.”

Thomas Edison